

## EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	359	(database near (dimension\$1 or multi-dimension or multidimension or two-dimension or (plurality adj2 dimension\$1)))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/24 18:19
L2	28	(database near (dimension\$1 or multi-dimension or multidimension or two-dimension or (plurality adj2 dimension\$1))) and (hierarch\$4 near2 attribute\$1)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/24 18:20
L3	28	(database near (dimension\$1 or multi-dimension or multidimension or two-dimension or (plurality adj2 dimension\$1))) and (hierarch\$4 near2 attribute\$1) and aggregat\$4	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/24 18:22
L4	34594	"707"/\$.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/24 18:23
L5	28	3 and 4	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/24 18:23
L6	12	((multi near dimension\$ or multidimensional) same database\$ same storage).ti.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/24 18:27

	Document ID	Kind Codes	Source	Issue Date	Pages
1	US 6721760 B1		USPAT	20040413	25
2	JP 2005141588 A		JPO	20050602	17
3	JP 11224261 A		JPO	19990817	9
4	US 20050071349 A		DERWENT	20050331	20
5	NZ 527535 A		DERWENT	20050324	1

	Title	Abstract
1	Method and apparatus for searching multi-dimensional database and storage medium stored search program of multi-dimensional database	
2	WORKFLOW MANAGEMENT SERVER WITH BUILT-IN MULTI-DIMENSIONAL DATABASE, MANAGEMENT METHOD THEREOF, STORAGE MEDIUM AND PROGRAM	
3	DATA STORAGE METHOD AND DATA RETRIEVAL METHOD FOR MULTI-DIMENSIONAL DATABASE	
4	Data indexing method in multidimensional database, involves reorganizing selected data cells in multi-dimensional data storage model based on data access information, to reduce time taken to access to selected cells	
5	Method of indexing data in multidimensional database for on-line analytical processing, involves reorganizing selected cells in multidimensional data storage model based on data access information generated based on user queries	

	Current OR	Current XRef	Retrieval Classif	Inventor
1	707/104.1			Ono; Akinori et al.
2				NISHIDA, KOJI
3				SUGA, MASATAKA et al.
4				JORDAN, P M et al.
5				JORDAN, P M et al.

	Document ID	Kind Codes	Source	Issue Date	Pages
6	WO 200180056 A	A2, A3	DERWENT	20011025	42
7	US 6289352 B		DERWENT	20010911	17
8	US 20010000536 A		DERWENT	20010426	53
9	US 6408292 B		DERWENT	20020618	62

	Title	Abstract
6	Communicating between database and remote site over a network e.g. for electronic screen displays and electronic data storage and retrieval, generates screen displays using scalable, web-based, multi-dimensional dynamic database	
7	Online analytical processing method for computer database, involves representing viewed information within multidimensional storage for referencing compound structure to obtain viewed information	
8	Sequence of records (tuples) storage system for multi-dimensional data and/or database management, using attribute value and cardinality store information to group and sort data	
9	Multidimensional database elements accessing system for on-line analytical processing, maps integer encoded storage address within memory associated with multidimensional database against MDB dimensions	

	Current OR	Current XRef	Retrieval Classif	Inventor
6				POINDEXTER, K S et al.
7				PROCTOR, A C
8				TARIN, S A
9				BAKALASH, R et al.

	Document ID	Kind Codes	Source	Issue Date	Pages
10	JP 2001022621 A		DERWENT	20010126	11
11	US 5592666 A		DERWENT	19970107	25
12	US 5319777 A		DERWENT	19940607	13



	<b>Title</b>	<b>Abstract</b>
10	Multidimensional database management system for computer, distributes data blocks in storage area of database, so that number of cells in each storage area is approximately equal	
11	Storing and retrieving multi-dimensional array from computer database e.g. for spreadsheet - stores index to identify element of array's dimension to which database-pointer corresponds, and index within sub-storage locations indicating if database pointer is in storage node corresp. to associated element	
12	Information storage and retrieval from multidimensional array - providing electronic spreadsheets connected through local area network and separate from databases upon which they operate	

	Current OR	Current XRef	Retrieval Classif	Inventor
10				
11				PEREZ, M
12				PEREZ, M

DERWENT-ACC-NO: 2001-206791

DERWENT-WEEK: 200121

COPYRIGHT 2006 DERWENT INFORMATION LTD

TITLE: Multidimensional database management system  
for  
computer, distributes data blocks in storage  
area of  
database, so that number of cells in each  
storage area is  
approximately equal

PATENT-ASSIGNEE: HITACHI LTD[HITA]

PRIORITY-DATA: 1999JP-0194984 (July 8, 1999)

PATENT-FAMILY:

PUB-NO	PUB-DATE	LANGUAGE
PAGES MAIN-IPC		
JP 2001022621 A	January 26, 2001	N/A
011 G06F 012/00		

APPLICATION-DATA:

PUB-NO	APPL-DESCRIPTOR	APPL-NO
APPL-DATE		
JP2001022621A	N/A	1999JP-0194984
July 8, 1999		

INT-CL (IPC): G06F012/00, G06F017/30

ABSTRACTED-PUB-NO: JP2001022621A

BASIC-ABSTRACT:

NOVELTY - Process node (202) processes data cells of different dimensions.  
Based on the number of storing areas of the database, input data is divided into data blocks and coordinate value is assigned to each block containing the cells. The data blocks are distributed in the storing areas of database, so that the number of data cells in each area is approximately equal.

USE - For multidimensional database management for computer.

ADVANTAGE - Parallel processing and calculation velocity are improved

by  
enabling suitable storage control of the data cells in the database.

DESCRIPTION OF DRAWING(S) - The figure shows the block diagram of  
database  
management system.

Process node 202

CHOSEN-DRAWING: Dwg.2/12

TITLE-TERMS: MULTIDIMENSIONAL DATABASE MANAGEMENT SYSTEM COMPUTER  
DISTRIBUTE

DATA BLOCK STORAGE AREA DATABASE SO NUMBER CELL STORAGE  
AREA

APPROXIMATE EQUAL

DERWENT-CLASS: T01

EPI-CODES: T01-H01; T01-J05B2;

SECONDARY-ACC-NO:

Non-CPI Secondary Accession Numbers: N2001-148028

DERWENT-ACC-NO: 2001-432376

DERWENT-WEEK: 200552

COPYRIGHT 2006 DERWENT INFORMATION LTD

TITLE: Multidimensional database elements accessing  
system for on-line analytical processing, maps integer  
encoded storage address within memory associated with  
multidimensional database against MDB  
dimensions

INVENTOR: BAKALASH, R; SHAKED, G

PATENT-ASSIGNEE: HYPERROLL ISRAEL LTD [HYPEN]

PRIORITY-DATA: 1999US-0368241 (August 4, 1999)

PATENT-FAMILY:

PUB-NO	PUB-DATE	LANGUAGE
PAGES MAIN-IPC		
WO 200111497 A1	February 15, 2001	E
097 G06F 017/30		
AU 200060108 A	March 5, 2001	N/A
000 G06F 017/30		
US 6408292 B1	June 18, 2002	N/A
000 G06F 017/30		

DESIGNATED-STATES: AE AG AL AM AT AU AZ BA BB BG BR BY BZ CA CH CN CR  
CU CZ DE  
DK DM DZ EE ES FI GB GD GE GH GM HR HU ID IL IN IS JP KE KG KP KR KZ  
LC LK LR  
LS LT LU LV MA MD MG MK MN MW MX MZ NO NZ PL PT RO RU SD SE SG SI SK  
SL TJ TM  
TR TT TZ UA UG US UZ VN YU ZA ZW AT BE CH CY DE DK EA ES FI FR GB GH  
GM GR IE  
IT KE LS LU MC MW MZ NL OA PT SD SE SL SZ TZ UG ZW

APPLICATION-DATA:

PUB-NO	APPL-DESCRIPTOR	APPL-NO
APPL-DATE		
WO 200111497A1	N/A	2000WO-IB01100
August 4, 2000		
AU 200060108A	N/A	2000AU-0060108
August 4, 2000		
AU 200060108A	Based on	WO 200111497
N/A		

US 6408292B1  
August 4, 1999

N/A

1999US-0368241

INT-CL (IPC): G06F017/30

RELATED-ACC-NO: 2001-625902, 2002-414155 , 2003-039626 , 2003-039629  
, 2003-058087 , 2003-220403 , 2003-902144 , 2004-060691 , 2004-061065  
, 2005-231954 , 2005-241102 , 2005-252860 , 2005-353444 , 2005-512094

ABSTRACTED-PUB-NO: US 6408292B

BASIC-ABSTRACT:

NOVELTY - Each data location in multidimensional database (MDB) is specified by integer encoded business dimensions associated with data. Address data mapping unit maps integer coded MDB dimensions against integer encoded data storage address within memory associated with MDB using modular arithmetic function. Data accessing unit accesses data element in memory using map information.

DETAILED DESCRIPTION - A parallel computing platform has processors and memories for storing data elements in integer encoded address.  
INDEPENDENT

CLAIMS are also included for the following:

- (a) Data element accessing method;
- (b) Data element management system;
- (c) Data element management method;
- (d) Internet URL directory system;
- (e) Internet enabled system

USE - For accessing multidimensional database (MDB) such as data warehouse in business organization for on-line analytical processing, MDB is used in on-line e-commerce shopping system for storing consumer shopping profile information, for URL directory system used for data mixing in Internet, and other MDB based system used for predictive business modeling for applications such as

database  
marketing, financial/risk analysis, fraud management, bioinformatics,  
return-on-investment justification, business intelligence  
application, customer  
relation management, enterprise information portals and systems used  
for  
supporting real-time control of packet routers, switches and other  
devices used  
in Internet, for real-time control of automated parcel routing and  
sortation  
system.

ADVANTAGE - Improved data accessing is provided by parallel computing  
platform.  
Inter process communication among parallel processors is minimized.  
Fast,  
affordable and easy access is provided to customer enabling companies  
to more  
effectively market products and service over internet. Supporting  
real-time  
control of processor in response to complex states of information  
reflected in  
MDB.

DESCRIPTION OF DRAWING(S) - The figure shows the schematic  
representation of  
data element address assignment method.

ABSTRACTED-PUB-NO: WO 200111497A

EQUIVALENT-ABSTRACTS:

NOVELTY - Each data location in multidimensional database (MDB) is  
specified by  
integer encoded business dimensions associated with data. Address  
data mapping  
unit maps integer coded MDB dimensions against integer encoded data  
storage  
address within memory associated with MDB using modular arithmetic  
function.  
Data accessing unit accesses data element in memory using map  
information.

DETAILED DESCRIPTION - A parallel computing platform has processors  
and  
memories for storing data elements in integer encoded address.  
INDEPENDENT  
CLAIMS are also included for the following:

(a) Data element accessing method;

- (b) Data element management system;
- (c) Data element management method;
- (d) Internet URL directory system;
- (e) Internet enabled system

USE - For accessing multidimensional database (MDB) such as data warehouse in business organization for on-line analytical processing, MDB is used in on-line e-commerce shopping system for storing consumer shopping profile information, for URL directory system used for data mixing in Internet, and other MDB based system used for predictive business modeling for applications such as database marketing, financial/risk analysis, fraud management, bioinformatics, return-on-investment justification, business intelligence application, customer relation management, enterprise information portals and systems used for supporting real-time control of packet routers, switches and other devices used in Internet, for real-time control of automated parcel routing and sortation system.

ADVANTAGE - Improved data accessing is provided by parallel computing platform. Inter process communication among parallel processors is minimized. Fast, affordable and easy access is provided to customer enabling companies to more effectively market products and service over internet. Supporting real-time control of processor in response to complex states of information reflected in MDB.

DESCRIPTION OF DRAWING(S) - The figure shows the schematic representation of data element address assignment method.

CHOSEN-DRAWING: Dwg.8A/18

TITLE-TERMS: MULTIDIMENSIONAL DATABASE ELEMENT ACCESS SYSTEM LINE ANALYSE



PROCESS MAP INTEGER ENCODE STORAGE ADDRESS MEMORY  
ASSOCIATE  
MULTIDIMENSIONAL DATABASE DIMENSION

DERWENT-CLASS: T01 W01

EPI-CODES: T01-D02; T01-H01A; T01-H07C5E; T01-J05A; T01-J05B4M; T01-M02C;  
W01-A03B; W01-A06B7; W01-A06E1; W01-A06G2;

SECONDARY-ACC-NO:

Non-CPI Secondary Accession Numbers: N2001-320438

DERWENT-ACC-NO: 2005-672796

DERWENT-WEEK: 200569

COPYRIGHT 2006 DERWENT INFORMATION LTD

TITLE: Method of indexing data in multidimensional database for on-line analytical processing, involves reorganizing selected cells in multidimensional data storage model based on data access information generated based on user queries

INVENTOR: JORDAN, P M; NG, K S ; SANDERS, M J ; STEWART, J

PATENT-ASSIGNEE: DESCISYS LTD[DESCN]

PRIORITY-DATA: 2003NZ-0527535 (August 11, 2003)

PATENT-FAMILY:

PUB-NO	PUB-DATE	LANGUAGE
PAGES MAIN-IPC		
NZ 527535 A	March 24, 2005	N/A
037 G06F 017/30		

APPLICATION-DATA:

PUB-NO	APPL-DESCRIPTOR	APPL-NO
APPL-DATE		
NZ 527535A	N/A	2003NZ-0527535
August 11, 2003		
NZ 527535A	Div in	NZ 537745
N/A		

INT-CL (IPC): G06F017/30, G06F017/40 , G06F017/60

ABSTRACTED-PUB-NO: NZ 527535A

BASIC-ABSTRACT:

NOVELTY - A multidimensional logical access model and a multidimensional data storage model are created. The data access information are generated based on user queries of the database. The selected cells in the multidimensional data storage model are reorganized based on the data access information to

reduce  
the cost of access to selected cells in response to user query of the  
database.

DETAILED DESCRIPTION - INDEPENDENT CLAIMS are also included for the  
following:

- (1) computerized apparatus for indexing data in multidimensional  
database; and
- (2) computer program for indexing data in multidimensional database.

USE - For indexing data in multidimensional database used in  
multidimensional  
on-line analytical processing (MOLAP) and relational OLAP (ROLAP).

ADVANTAGE - Optimizes data access by separating logical access model  
from data  
storage. Allows the database management software to automatically  
reorganize  
data in response to queries, and hence improves database performance.

DESCRIPTION OF DRAWING(S) - The figure shows a data structure of a  
multidimensional database model.

CHOSEN-DRAWING: Dwg.1/9

TITLE-TERMS: METHOD INDEX DATA MULTIDIMENSIONAL DATABASE LINE ANALYSE  
PROCESS

SELECT CELL MULTIDIMENSIONAL DATA STORAGE MODEL BASED  
DATA ACCESS  
INFORMATION GENERATE BASED USER QUERY

DERWENT-CLASS: T01

EPI-CODES: T01-J05B1; T01-J05B4M;

SECONDARY-ACC-NO:

Non-CPI Secondary Accession Numbers: N2005-551701



Welcome United States Patent and Trademark Office

Search Results

[BROWSE](#)[SEARCH](#)[IEEE XPLORE GUIDE](#)[SUPPORT](#)

Results for "((((multidimensional &lt;near/3&gt; database)&lt;in&gt;metadata))&lt;and&gt;((multidimensional &lt;..."

Your search matched 7 of 13 documents.

A maximum of 100 results are displayed, 25 to a page, sorted by **Relevance** in **Descending** order.

e-mail
 printer friendly

## » Search Options

[View Session History](#)[New Search](#)

## » Key

IEEE JNL	IEEE Journal or Magazine
IEE JNL	IEE Journal or Magazine
IEEE CNF	IEEE Conference Proceeding
IEE CNF	IEE Conference Proceeding
IEEE STD	IEEE Standard

## Modify Search


☐ Check to search only within this results set

 Display Format:
 ☒ Citation
 ☐ Citation & Abstract

[Select All](#)
[Deselect All](#)

- ☐
**1. Designing data warehouses with OO conceptual models**  
 Trujillo, J.; Palomar, M.; Gomez, J.; Il-Yeol Song;  
Computer  
 Volume 34, Issue 12, Dec. 2001 Page(s):66 - 75  
 Digital Object Identifier 10.1109/2.970579  
[AbstractPlus](#) | [References](#) | Full Text: [PDF](#)(989 KB) IEEE JNL  
[Rights and Permissions](#)
  
- ☐
**2. Summary grids: building accurate multidimensional histograms**  
 Furtado, P.; Madeira, H.;  
Database Systems for Advanced Applications, 1999. Proceedings, 6th International Conference on  
 19-21 April 1999 Page(s):187 - 194  
 Digital Object Identifier 10.1109/DASFAA.1999.765751  
[AbstractPlus](#) | Full Text: [PDF](#)(144 KB) IEEE CNF  
[Rights and Permissions](#)
  
- ☐
**3. Modeling multidimensional databases**  
 Agrawal, R.; Gupta, A.; Sarawagi, S.;  
Data Engineering, 1997. Proceedings, 13th International Conference on  
 7-11 April 1997 Page(s):232 - 243  
 Digital Object Identifier 10.1109/ICDE.1997.581777  
[AbstractPlus](#) | Full Text: [PDF](#)(972 KB) IEEE CNF  
[Rights and Permissions](#)
  
- ☐
**4. Finding your way through multidimensional data models**  
 Blaschka, M.; Sapia, C.; Hofling, G.; Dinter, B.;  
Database and Expert Systems Applications, 1998. Proceedings, Ninth International Workshop on  
 26-28 Aug. 1998 Page(s):198 - 203  
 Digital Object Identifier 10.1109/DEXA.1998.707403  
[AbstractPlus](#) | Full Text: [PDF](#)(72 KB) IEEE CNF  
[Rights and Permissions](#)
  
- ☐
**5. Partial-sum queries in OLAP data cubes using covering codes**  
 Ching-Tien Ho; Bruck, J.; Agrawal, R.;  
Computers, IEEE Transactions on  
 Volume 47, Issue 12, Dec. 1998 Page(s):1326 - 1340  
 Digital Object Identifier 10.1109/12.737680  
[AbstractPlus](#) | [References](#) | Full Text: [PDF](#)(568 KB) IEEE JNL  
[Rights and Permissions](#)

- ☐ 6. **Maintaining data cubes under dimension updates**  
Hurtado, C.A.; Mendelzon, A.O.; Vaisman, A.A.;  
[Data Engineering, 1999. Proceedings., 15th International Conference on](#)  
23-26 March 1999 Page(s):346 - 355  
Digital Object Identifier 10.1109/ICDE.1999.754950  
[AbstractPlus](#) | [Full Text: PDF\(304 KB\)](#) IEEE CNF  
[Rights and Permissions](#)
- ☐ 7. **Polaris: a system for query, analysis, and visualization of multidimensional relational databases**  
Stolte, C.; Tang, D.; Hanrahan, P.;  
[Visualization and Computer Graphics, IEEE Transactions on](#)  
Volume 8, Issue 1, Jan.-March 2002 Page(s):52 - 65  
Digital Object Identifier 10.1109/2945.981851  
[AbstractPlus](#) | [References](#) | [Full Text: PDF\(3273 KB\)](#) IEEE JNL  
[Rights and Permissions](#)

Indexed by  
 Inspec®

[Help](#) [Contact Us](#) [Privacy & Security](#) [IEEE.org](#)  
© Copyright 2006 IEEE – All Rights Reserved


[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☒ The ACM Digital Library ☐ The Guide

multidimensional and database and attribute and aggregate

SEARCH

THE ACM DIGITAL LIBRARY

[Feedback](#) [Report a problem](#) [Satisfaction survey](#)

Terms used

[multidimensional](#) and [database](#) and [attribute](#) and [aggregate](#)

 Found **22,852** of **176,279**

 Sort results  
by

relevance

☒ Save results to a Binder

 Try an [Advanced Search](#)

 Display  
results

expanded form

☒ Search Tips

 Try this search in [The ACM Guide](#)
☐ Open results in a new window

Results 1 - 20 of 200

 Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

 Relevance scale ☐ ☐ ☐ ☐ ☐

### 1 [Heuristic optimization of OLAP queries in multidimensionally hierarchically clustered](#)



#### [databases](#)

Dimitri Theodoratos, Aris Tsois

 November 2001 **Proceedings of the 4th ACM international workshop on Data warehousing and OLAP**

Publisher: ACM Press

Full text available: pdf(1.44 MB)

 Additional Information: [full citation](#), [abstract](#), [citations](#), [index terms](#)

On-line analytical processing (OLAP) is a technology that encompasses applications requiring a multidimensional and hierarchical view of data. OLAP applications often require fast response time to complex grouping/aggregation queries on enormous quantities of data. Commercial relational database management systems use mainly multiple one-dimensional indexes to process OLAP queries that restrict multiple dimensions. However, in many cases, multidimensional access methods outperform one-dimensiona ...

### 2 [Query processing: Exploiting hierarchical clustering in evaluating multidimensional aggregation queries](#)



#### [aggregation queries](#)

Dimitri Theodoratos

 November 2003 **Proceedings of the 6th ACM international workshop on Data warehousing and OLAP**

Publisher: ACM Press

Full text available: pdf(216.79 KB)

 Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Multidimensional aggregation queries constitute the single most important class of queries for data warehousing applications and decision support systems. The bottleneck in the evaluation of these queries is the join of the usually huge fact table with the restricted dimension tables (*star-join*). Recently, a multidimensional hierarchical clustering schema for star schemas is suggested. Subsequently, query evaluation plans for multidimensional queries appeared that essentially implement a ...

**Keywords:** multidimensional aggregation query, multidimensional hierarchical clustering, query transformations, star join

### 3 [Understanding facts in a multidimensional object-oriented model](#)



#### [Alberto Abelló, José Samos, Félix Saltor](#)

 November 2001 **Proceedings of the 4th ACM international workshop on Data warehousing and OLAP**

Publisher: ACM Press

Full text available: pdf(1.37 MB)

 Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index](#)

terms

"On-Line Analytical Processing" tools are used to extract information from the "Data Warehouse" in order to help in the decision making process. These tools are based on multidimensional concepts, i.e. facts and dimensions. In this paper we study the meaning of facts, and the dependencies in multidimensional data. This study is used to find relationships between cubes (in an Object-Oriented framework) and explain navigation operations.

**Keywords:** cube operations, functional dependencies, multidimensionality, object-oriented modeling

#### 4 Comparing nested GPSJ queries in multidimensional databases



Matteo Golfarelli, Stefano Rizzi

November 2000 **Proceedings of the 3rd ACM international workshop on Data warehousing and OLAP**

**Publisher:** ACM Press

Full text available: [pdf\(244.04 KB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

#### 5 Selectivity estimators for multidimensional range queries over real attributes

Dimitrios Gunopulos, George Kollios, J. Tsotras, Carlotta Domeniconi

April 2005 **The VLDB Journal — The International Journal on Very Large Data Bases**, Volume 14 Issue 2

**Publisher:** Springer-Verlag New York, Inc.

Full text available: [pdf\(321.96 KB\)](#) Additional Information: [full citation](#), [abstract](#)

Estimating the selectivity of multidimensional range queries over real valued attributes has significant applications in data exploration and database query optimization. In this paper, we consider the following problem: given a table of  $d$  attributes whose domain is the real numbers and a query that specifies a range in each dimension, find a good approximation of the number of records in the table that satisfy the query. The simplest approach to tackle this problem is to assume that the ...

#### 6 A survey of logical models for OLAP databases



Panos Vassiliadis, Timos Sellis

December 1999 **ACM SIGMOD Record**, Volume 28 Issue 4

**Publisher:** ACM Press

Full text available: [pdf\(604.36 KB\)](#) Additional Information: [full citation](#), [abstract](#), [citations](#), [index terms](#)

In this paper, we present different proposals for multidimensional data cubes, which are the basic logical model for OLAP applications. We have grouped the work in the field in two categories: commercial tools (presented along with terminology and standards) and academic efforts. We further divide the academic efforts in two subcategories: the relational model extensions and the cube-oriented approaches. Finally, we attempt a comparative analysis of the various efforts.

#### 7 Range queries in OLAP data cubes



Ching-Tien Ho, Rakesh Agrawal, Nimrod Megiddo, Ramakrishnan Srikant

June 1997 **ACM SIGMOD Record , Proceedings of the 1997 ACM SIGMOD international conference on Management of data SIGMOD '97**, Volume 26 Issue 2

**Publisher:** ACM Press

Full text available: [pdf\(1.91 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

A range query applies an aggregation operation over all selected cells of an OLAP data cube where the selection is specified by providing ranges of values for numeric dimensions. We present fast algorithms for range queries for two types of aggregation operations: SUM and MAX. These two operations cover techniques required for most popular aggregation operations, such as those supported by SQL. For range-sum queries,

the essential idea is to precompute some auxiliary information ...

8 OLAP and statistical databases: similarities and differences



Arie Shoshani

May 1997 **Proceedings of the sixteenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems**

**Publisher:** ACM Press

Full text available: pdf(1.66 MB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

9 Towards on-line analytical mining in large databases



Jiawei Han

March 1998 **ACM SIGMOD Record**, Volume 27 Issue 1

**Publisher:** ACM Press

Full text available: pdf(387.04 KB) Additional Information: [full citation](#), [abstract](#), [citations](#), [index terms](#)

Great efforts have been paid in the Intelligent Database Systems Research Lab for the research and development of efficient data mining methods and construction of on-line analytical data mining systems. Our work has been focused on the integration of data mining and OLAP technologies and the development of scalable, integrated, and multiple data mining functions. A data mining system, DBMiner, has been developed for interactive mining of multiple-level knowledge in large relational databases and ...

10 High performance multidimensional analysis of large datasets



Sanjay Goil, Alok Choudhary

November 1998 **Proceedings of the 1st ACM international workshop on Data warehousing and OLAP**

**Publisher:** ACM Press

Full text available: pdf(856.78 KB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

11 Data mining and database technology: A new histogram method for sparse attributes: the averaged rectangular attribute cardinality map



B. John Oommen, Jing Chen

September 2003 **Proceedings of the 1st international symposium on Information and communication technologies ISICT '03**

**Publisher:** Trinity College Dublin

Full text available: pdf(153.70 KB) Additional Information: [full citation](#), [abstract](#), [references](#)

Most current Database Management Systems (DBMS) use histograms in their query optimization, and in approximating query result sizes. This is because they can be utilized in determining efficient query evaluation plans. All the existing methods perform poorly when the attributes of a relation are very sparsely distributed, also called the "sparse data cases". These cases are the worst-cases scenarios for attributes with skewed distributions. In this paper, we propose a novel histogram-based algor ...

**Keywords:** attribute cardinality maps, histogram-based algorithms, query optimization, sparse attributes

12 Research sessions: data mining applications: Diamond in the rough: finding



Hierarchical Heavy Hitters in multi-dimensional data

Graham Cormode, Flip Korn, S. Muthukrishnan, Divesh Srivastava

June 2004 **Proceedings of the 2004 ACM SIGMOD international conference on Management of data**

**Publisher:** ACM Press

Full text available: pdf(372.53 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)



Data items archived in data warehouses or those that arrive online as streams typically have attributes which take values from multiple hierarchies (e.g., time and geographic location; source and destination IP addresses). Providing an aggregate view of such data is important to summarize, visualize, and analyze. We develop the aggregate view based on certain hierarchically organized sets of large-valued regions ("heavy hitters"). Such Hierarchical Heavy Hitters (HHHs) were previously introduced ...

### 13 Research session 4: query processing and optimization I: How to evaluate multiple range-sum queries progressively



Rolfe R. Schmidt, Cyrus Shahabi

June 2002 **Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems**

**Publisher:** ACM Press

Full text available: [pdf\(2.14 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Users of decision support system typically submit batches of range-sum queries simultaneously rather than issuing individual, unrelated queries. We propose a wavelet based technique that exploits T/O sharing across a query batch to evaluate the set of queries progressively and efficiently. The challenge is that now controlling the structure of errors across query results becomes more critical than minimizing error per individual query. Consequently, we define a class of structural error penalty ...

### 14 Complete answer aggregates for treelike databases: a novel approach to combine querying and navigation



Holger Meuss, Klaus U. Schulz

April 2001 **ACM Transactions on Information Systems (TOIS)**, Volume 19 Issue 2

**Publisher:** ACM Press

Full text available: [pdf\(356.60 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

The use of markup languages like SGML, HTML or XML for encoding the structure of documents or linguistic data has lead to many databases where entries are adequately described as trees. In this context querying formalisms are interesting that offer the possibility to refer both to textual content and logical structure. We consider models where the structure specified in a query is not only used as a filter, but also for selecting and presenting different parts of the data. If answers are formaliz ...

**Keywords:** SGML, XML, answer presentation, information retrieval, logic, query languages, semistructured data, structured documents, tree databases, tree matching

### 15 Top-k selection queries over relational databases: Mapping strategies and performance evaluation



Nicolas Bruno, Surajit Chaudhuri, Luis Gravano

June 2002 **ACM Transactions on Database Systems (TODS)**, Volume 27 Issue 2

**Publisher:** ACM Press

Full text available: [pdf\(1.64 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

In many applications, users specify target values for certain attributes, without requiring exact matches to these values in return. Instead, the result to such queries is typically a rank of the "top k" tuples that best match the given attribute values. In this paper, we study the advantages and limitations of processing a top-k query by translating it into a single range query that a traditional relational database management system (RDBMS) can process efficiently. In particular, ...

**Keywords:** Multidimensional histograms, top-k query processing

**16** Synopses for query optimization: A space-complexity perspective

 Raghav Kaushik, Jeffrey F. Naughton, Raghu Ramakrishnan, Venkatesan T. Chakravarthy  
December 2005 **ACM Transactions on Database Systems (TODS)**, Volume 30 Issue 4


**Publisher:** ACM Press

Full text available:  pdf(530.68 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Database systems use precomputed synopses of data to estimate the cost of alternative plans during query optimization. A number of alternative synopsis structures have been proposed, but histograms are by far the most commonly used. While histograms have proved to be very effective in (cost estimation for) single-table selections, queries with joins have long been seen as a challenge; under a model where histograms are maintained for individual tables, a celebrated result of Ioannidis and Christ ...

**Keywords:** Cardinality estimation, histograms, sampling

**17** A dynamic load balancing strategy for parallel datacube computation

 Seigo Muto, Masaru Kitsuregawa  
November 1999 **Proceedings of the 2nd ACM international workshop on Data warehousing and OLAP**

**Publisher:** ACM Press

Full text available:  pdf(732.24 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)


In recent years, OLAP technologies have become one of the important applications in the database industry. In particular, the datacube operation proposed in [5] receives strong attention among researchers as a fundamental research topic in the OLAP technologies. The datacube operation requires computation of aggregations on all possible combinations of each dimension attribute. As the number of dimensions increases, it becomes very expensive to compute datacubes, because the required comput ...

**Keywords:** OLAP, datacube, dynamic load balancing, parallel processing

**18** Browsing large digital library collections using classification hierarchies

 S. Geffner, D. Agrawal, A. El Abbadi, T. Smith  
November 1999 **Proceedings of the eighth international conference on Information and knowledge management**


**Publisher:** ACM Press

Full text available:  pdf(998.73 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Summarization of intermediary query result sets plays an important role when users browse through digital library collections. Summarization enables users to quickly digest the results of their queries, and provides users with important information they can use to narrow their search interactively. Techniques from the field of data analysis may be applied to the problem of generating summaries of query results efficiently. Such techniques should permit the incorporation of classification hi ...

**Keywords:** aggregation, browsing, classification, digital libraries, searching, summarization

**19** Security-control methods for statistical databases: a comparative study

 Nabil R. Adam, John C. Worthmann  
December 1989 **ACM Computing Surveys (CSUR)**, Volume 21 Issue 4

**Publisher:** ACM Press

Full text available:  pdf(3.64 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

This paper considers the problem of providing security to statistical databases against disclosure of confidential information. Security-control methods suggested in the literature are classified into four general approaches: conceptual, query restriction, data perturbation, and output perturbation. Criteria for evaluating the performance of the various security-control methods are identified. Security-control methods that are based on each of the four approaches are discussed, t ...

## 20 Extending OLAP querying to external object databases

 Torben Bach Pedersen, Arie Shoshani, Junmin Gu, Christian S. Jensen

November 2000 **Proceedings of the ninth international conference on Information and knowledge management**

**Publisher:** ACM Press

Full text available:  [pdf\(168.32 KB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2006 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)

[Web](#) [Images](#) [Groups](#) [News](#) [Froogle](#) [Maps](#) [more »](#)

storing data values multidimensional database

[Advanced Search](#)  
[Preferences](#)**Web** Results 1 - 10 of about 81,700 for storing data values multidimensional database hierarchy attribute aggregate. (0.51

### OLAP Glossaire

**Aggregate:** The act of calculating **values** associated with parent positions in the ...  
Designates **multidimensional** analysis tools which **store data** in either ...  
[perso.wanadoo.fr/bernard.lupin/english/glossary.htm](http://perso.wanadoo.fr/bernard.lupin/english/glossary.htm) - 15k - [Cached](#) - [Similar pages](#)

### 2 The Multidimensional Data Model

While **attributes** are often **multidimensional**, only one dimension is a **data** dimension. A **hierarchy** dimension, which lists the **data** dimension hierarchies, ...  
[www.stanford.edu/dept/itss/docs/oracle/10g/olap.101/b10333/multimodel.htm](http://www.stanford.edu/dept/itss/docs/oracle/10g/olap.101/b10333/multimodel.htm) - 38k - [Cached](#) - [Similar pages](#)

#### Glossary

In a dimension **hierarchy**, the **data value** of the ancestor is the aggregated **value** of the ...  
In an analytic workspace, **aggregate data** can be precalculated, ...  
[www.stanford.edu/dept/itss/docs/oracle/10g/olap.101/b10333/awgloss.htm](http://www.stanford.edu/dept/itss/docs/oracle/10g/olap.101/b10333/awgloss.htm) - 49k - [Cached](#) - [Similar pages](#)  
[ [More results from www.stanford.edu](#) ]

### Stöhr, Thomas; Märtens, Holger; Rahm, Erhard: Multi-Dimensional ...

**Multi-Dimensional Database** Allocation for Parallel **Data** Warehouses, Proc. ... A selection for a specific **attribute value** (eg, RefTo- Time= MONTH in query ...  
[lips.informatik.uni-leipzig.de/pub/2000-12](http://lips.informatik.uni-leipzig.de/pub/2000-12) - 70k - [Cached](#) - [Similar pages](#)

### Designing Data Warehouses with OO Conceptual Models

Most developers agree that **data** warehouse, **multidimensional database** (MDB), ...  
Applying a roll-up operation to **aggregate** measure **values** from the **store** ...  
[doi.ieeeecomputersociety.org/10.1109/2.970579](http://doi.ieeeecomputersociety.org/10.1109/2.970579) - [Similar pages](#)

### Iconium - Business Intelligence Glossary

A state in which all the **data values** stored in the **database** are correct. ... A structural **attribute** of a cube, which is an organised **hierarchy** of categories ...  
[www.iconium.co.uk/biglossary.htm](http://www.iconium.co.uk/biglossary.htm) - 56k - [Cached](#) - [Similar pages](#)

### Healthcare Informatics: Data Mining Goes Multidimensional

**Storing** summary **data** in **multidimensional** form reduces the need to perform **aggregate** calculations at run time during every analysis. Although its emphasis is ...  
[www.healthcare-informatics.com/issues/1997/03\\_97/sys\\_rev.htm](http://www.healthcare-informatics.com/issues/1997/03_97/sys_rev.htm) - 35k - [Cached](#) - [Similar pages](#)

### The OLAP Report: Database Explosion

**Multidimensional database** storage: Contrary to the propaganda by some vendors, the type of **database** technology used to **store multidimensional data** has ...  
[www.olapreport.com/DatabaseExplosion.htm](http://www.olapreport.com/DatabaseExplosion.htm) - 42k - [Cached](#) - [Similar pages](#)

### [PDF] Designing Data Warehouses with OO Conceptual Models

File Format: PDF/Adobe Acrobat - [View as HTML](#)  
ost developers agree that **data** warehouse, **multidimensional database** (MDB), and ... tor to **aggregate attribute values** along all hierarchies ...  
[www.dlsi.ua.es/~mpalomar/gold.pdf](http://www.dlsi.ua.es/~mpalomar/gold.pdf) - [Similar pages](#)

### Designing Your Database for OLAP, 6 of 6

In contrast, the basic **data** model for **multidimensional** analysis is a cube, which is

composed of ... Whereas level **attributes** map to specific **data values**, ...

[www.utexas.edu/its/unix/reference/](http://www.utexas.edu/its/unix/reference/)

[oracledocs/v92/B10501\\_01/olap.920/a95295/designd6.htm](http://oracledocs/v92/B10501_01/olap.920/a95295/designd6.htm) - 38k - [Cached](#) - [Similar pages](#)

**Google Groups results for storing data values multidimensional database hierarchy attribute aggregate**



[WANTED: Volume Rendering Bibliography](#) - comp.graphics.visualization - Apr 22, 1994

[Shell2 -- the outrigger \(2/4\) cgv.FAQ2](#) - comp.graphics.visualization - Oct 22, 2004

[Python Language FAQ - Section 4](#) - comp.lang.python.announce - Jul 07, 1999

Try your search again on [Google Book Search](#)

Goooooooooooooogle ►

Result Page:    1   2   3   4   5   6   7   8   9   10    **Next**

Free! Speed up the web. [Download the Google Web Accelerator.](#)

storing data values multidimensional

[Search within results](#) | [Language Tools](#) | [Search Tips](#) | [Dissatisfied? Help us improve](#)

[Google Home](#) - [Advertising Programs](#) - [Business Solutions](#) - [About Google](#)

©2006 Google

[Web](#) [Images](#) [Groups](#) [News](#) [Froogle](#) [Maps](#) [more »](#)

multidimensional database storage and retrieval

[Advanced Search](#)  
[Preferences](#)The "AND" operator is unnecessary -- we include all search terms by default. [\[details\]](#)**Web**Results 1 - 10 of about **476,000** for **multidimensional database storage and retrieval** . (0.42 seconds)[\[PDF\]](#) **HEAVEN A Hierarchical Storage and Archive Environment for ...**File Format: PDF/Adobe Acrobat - [View as HTML](#)**Multidimensional Array Database Management Systems ...** duce the common state of the art technique **storage and retrieval** of large spatio- ...[www.wibas.forwiss.tu-muenchen.de/wibas/](http://www.wibas.forwiss.tu-muenchen.de/wibas/db/webcon/wibasdb/download/51/application/pdf.pdf)[db/webcon/wibasdb/download/51/application/pdf.pdf](http://db/webcon/wibasdb/download/51/application/pdf.pdf) - [Similar pages](#)**Details****Multidimensional database** technology involves DBMS support wrt. ... suitable **storage** management, including tertiary **storage**; **storage and retrieval** ...[www.wibas.forwiss.tu-muenchen.de/wibas/](http://www.wibas.forwiss.tu-muenchen.de/wibas/db/webcon/wibasdb/details/Projekte/num/20/8.html)[db/webcon/wibasdb/details/Projekte/num/20/8.html](http://db/webcon/wibasdb/details/Projekte/num/20/8.html) - 9k - [Cached](#) - [Similar pages](#)**Database Research Group @ UCI**Multimedia Analysis and **Retrieval** System (MARS) ... Our research addresses the information **retrieval** and **database** management challenges in content-based ...[www-db.ics.uci.edu/pages/research/mars.shtml](http://www-db.ics.uci.edu/pages/research/mars.shtml) - 18k - [Cached](#) - [Similar pages](#)**Storage and Retrieval for Image and Video Databases 1995****Storage and Retrieval** for Image and Video Databases 1995: San Diego/La Jolla ... SimHeng Ong: Image **Retrieval** Based on **Multidimensional** Feature Properties. ...[www.informatik.uni-trier.de/~ley/db/conf/spieSR/spieSR95.html](http://www.informatik.uni-trier.de/~ley/db/conf/spieSR/spieSR95.html) - 17k -[Cached](#) - [Similar pages](#)**Storage and Retrieval for Image and Video Databases 1996****Storage and Retrieval** for Image and Video Databases 1996: San Diego/La Jolla, CA, ...Andishe Sedighian: Evaluating **Multidimensional** Indexing Structures for ...[www.informatik.uni-trier.de/~ley/db/conf/spieSR/spieSR96.html](http://www.informatik.uni-trier.de/~ley/db/conf/spieSR/spieSR96.html) - 18k -[Cached](#) - [Similar pages](#)**Release of TigerLogic XDMS**... the Company's time-proven hierarchical and **multi-dimensional** Pick® **database** ...scaleable and reliable **storage and retrieval** of critical business data ...[www.rainingdata.com/company/pressroom/ttrelease.html](http://www.rainingdata.com/company/pressroom/ttrelease.html) - 41k - [Cached](#) - [Similar pages](#)**XML Database, SOA Solutions, Asset Tracking, ePedigree Compliance ...**NET, multivalue **Database**, and Rapid Application Tools, ... scalable and extensible **storage and retrieval** of critical business data across a variety of ...[www.rainingdata.com/products/index.html](http://www.rainingdata.com/products/index.html) - 44k - [Cached](#) - [Similar pages](#)**ESTEDI**... the **storage and retrieval** of **multidimensional** high-performance computing (HPC) ...To this end, the **multidimensional database** system RasDaMan will be ...[www.e-science.clrc.ac.uk/web/projects/estedi](http://www.e-science.clrc.ac.uk/web/projects/estedi) - 15k - [Cached](#) - [Similar pages](#)[\[PDF\]](#) **Tertiary Storage Support for Large-Scale Multidimensional Array ...**File Format: PDF/Adobe Acrobat - [View as HTML](#)efficient **storage, retrieval** or manipulation of MDDs. ... **Multidimensional**. Arrays. in.**Database**. Management Systems, PhD Thesis of Technical ...[www.abayfor.de/fv/forwiss/downloads/B0000035.pdf](http://www.abayfor.de/fv/forwiss/downloads/B0000035.pdf) - [Similar pages](#)**Databases and SQL - Links999.**

[www.links999.net/reference/databases.html](http://www.links999.net/reference/databases.html) - 41k - Cached - Similar pages

Goooooooooooooogle ▶

Info when you want it, right on your desktop  
Free! Download Google Desktop



[Search within results](#) | [Language Tools](#) | [Search Tips](#) | [Dissatisfied? Help us improve](#)

©2006 Google


[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☒ The ACM Digital Library ☐ The Guide

 **SEARCH**
**THE ACM DIGITAL LIBRARY**
[Feedback](#) [Report a problem](#) [Satisfaction survey](#)

## Heuristic optimization of OLAP queries in multidimensionally hierarchically clustered databases

**Full text** [Pdf \(1.44 MB\)](#)

**Source** [Data Warehousing and OLAP](#) [archive](#)  
**Proceedings of the 4th ACM international workshop on Data warehousing and OLAP** [table of contents](#)  
 Atlanta, Georgia, USA  
 Pages: 48 - 55  
 Year of Publication: 2001  
 ISBN:1-58113-437-1

**Authors** [Dimitri Theodoratos](#) [New Jersey Institute of Technology, Newark, NJ](#)  
[Aris Tsois](#) [National Technical University of Athens, Athens, Greece](#)

**Sponsors** [SIGMIS](#): ACM Special Interest Group on Management Information Systems  
 ACM : Assoc. for Computing Machinery  
[SIGIR](#): ACM Special Interest Group on Information Retrieval

**Publisher** ACM Press New York, NY, USA

**Additional Information:** [abstract](#) [citations](#) [index terms](#) [collaborative colleagues](#)

**Tools and Actions:** [Discussions](#) [Find similar Articles](#) [Review this Article](#)  
[Save this Article to a Binder](#) [Display Formats:](#) [BibTex](#) [EndNote](#) [ACM Ref](#)

**DOI Bookmark:** Use this link to bookmark this Article: <http://doi.acm.org/10.1145/512236.512243>  
[What is a DOI?](#)

### ↑ ABSTRACT

On-line analytical processing (OLAP) is a technology that encompasses applications requiring a multidimensional and hierarchical view of data. OLAP applications often require fast response time to complex grouping/aggregation queries on enormous quantities of data. Commercial relational database management systems use mainly multiple one-dimensional indexes to process OLAP queries that restrict multiple dimensions. However, in many cases, multidimensional access methods outperform one-dimensional indexing methods. We present an architecture for multidimensional databases that are clustered with respect to multiple hierarchical dimensions. It is based on the star schema and is called CSB star. Then, we focus on heuristically optimizing OLAP queries over this schema using multidimensional access methods. Users can still formulate their queries over a traditional star scheme, which are then rewritten by the query processor over the CSB star. We exploit the different clustering features of the CSB star to efficiently process a class of typical OLAP queries. We detect special cases where the construction of an evaluation plan can be simplified and we discuss improvements of our technique.

### ↑ CITINGS 2

[Dimitri Theodoratos, Exploiting hierarchical clustering in evaluating multidimensional aggregation queries, Proceedings of the 6th ACM international workshop on Data warehousing and OLAP, November 07-07, 2003, New Orleans, Louisiana, USA](#)

[Dimitri Theodoratos , Aris Tsois, Processing OLAP queries in hierarchically clustered databases, Data](#)



& Knowledge Engineering, v.45 n.2, p.205-224, May 2003

## ↑ INDEX TERMS

### Primary Classification:

**H.** [Information Systems](#)

↳ **H.2** [DATABASE MANAGEMENT](#)

↳ **H.2.4** [Systems](#)

↳ **Subjects:** [Query processing](#)

### Additional Classification:

**G.** [Mathematics of Computing](#)

↳ **G.1** [NUMERICAL ANALYSIS](#)

↳ **G.1.6** [Optimization](#)

**H.** [Information Systems](#)

↳ **H.2** [DATABASE MANAGEMENT](#)

↳ **H.2.4** [Systems](#)

↳ **Subjects:** [Distributed databases](#); [Relational databases](#)

**I.** [Computing Methodologies](#)

↳ **I.2** [ARTIFICIAL INTELLIGENCE](#)

↳ **I.2.8** [Problem Solving, Control Methods, and Search](#)

↳ **Subjects:** [Heuristic methods](#)

### General Terms:

[Algorithms](#), [Design](#), [Theory](#)

## ↑ Collaborative Colleagues:

[Dimitri Theodoratos](#): [Mokrane Bouzeghoub](#) [Manos Stavropoulos](#)

[Theodore Dalamagas](#) [Aris Tsois](#)

[Narain Gehani](#) [Yannis Vassiliou](#)

[Antonis Koufopoulos](#) [Wugang Xu](#)

[Spyros Ligoudistianos](#)

[Vincent Oria](#)

[Timos Sellis](#)

[Timos K. Sellis](#)

[Alkis Simitsis](#)

[Il-Yeol Song](#)

[Aris Tsois](#):

[Verena Kantere](#)

[Dimitri Theodoratos](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2006 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)

# Heuristic Optimization of OLAP Queries in Multidimensionally Hierarchically Clustered Databases\*

Dimitri Theodoratos  
Department of Computer Science  
New Jersey Institute of Technology  
University Heights, Newark, NJ 07102, USA  
dth @cis.njit.edu

Aris Tsois  
Dept. of Electrical and Computer Engineering  
National Technical University of Athens  
Zographou 157 73, Athens, Greece  
atsois@dblab.ece.ntua.gr

## ABSTRACT

On-Line Analytical Processing (OLAP) is a technology that encompasses applications requiring a multidimensional and hierarchical view of data. OLAP applications often require fast response time to complex grouping/aggregation queries on enormous quantities of data. Commercial relational database management systems use mainly multiple one-dimensional indexes to process OLAP queries that restrict multiple dimensions. However, in many cases, multidimensional access methods outperform one-dimensional indexing methods.

We present an architecture for multidimensional databases that are clustered with respect to multiple hierarchical dimensions. It is based on the star schema and is called CSB star. Then, we focus on heuristically optimizing OLAP queries over this schema using multidimensional access methods. Users can still formulate their queries over a traditional star schema, which are then rewritten by the query processor over the CSB star. We exploit the different clustering features of the CSB star to efficiently process a class of typical OLAP queries. We detect special cases where the construction of an evaluation plan can be simplified and we discuss improvements of our technique.

## 1. INTRODUCTION

Decision support applications increasingly rely on On-Line Analytical Processing (OLAP) to analyze business related information. OLAP is a technology that encom-

passes applications requiring a multidimensional view of data. In such a view of data there is a set of measures that are the metrics of interest. The measures contain numeric data. Each of them is uniquely determined by a set of different and often independent dimensions. Dimensions have associated with them hierarchies that specify different aggregation levels of data and hence different granularities of viewing data. Many relational OLAP systems use the star schema [12] to represent the multidimensional data model. A multidimensional database organized as a star consists of a fact table and a table for each dimension. A dimension table comprises attributes for each (aggregation) level of the dimension and other (descriptive) attributes that characterize the different levels of the dimension. The fact table stores attributes for each numeric measure and foreign key attributes to the attribute of the finest granularity level of each dimension.

OLAP applications often require fast response time to complex grouping/aggregation queries on enormous quantities of data. Common techniques to improve query performance are materializing views, and making extensive use of clustering and indexing methods. In multidimensional databases these techniques have to be adapted in order to account for the multiple dimensions. Materializing views is an efficient technique when the way to compute a query using the materialized views is known. However, the general problem of answering and optimizing grouping/aggregation queries using multiple materialized views [4, 3, 20] is complex. Another difficulty of the view materializing technique is the optimal selection of views to materialize [11, 19]. View selection becomes even more complex when the query pattern is not known in advance [13]. Lastly, this technique incurs important additional space requirements and intricate algorithms for incrementally maintaining the materialized views [9].

Commercial relational database management systems use mainly multiple one-dimensional indexes, like compound indexes and bitmap indexes, to process OLAP queries that restrict multiple dimensions. The search key in compound indexes is a concatenation of multiple attributes (where the order of attributes matter). Therefore, they are useful for processing only some of the queries that restrict these attributes. Selecting views

---

\*Research supported by the European Commission under the IST Program project EDITH: European Development of Indexing Techniques for Databases with Multidimensional Hierarchies

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

DOLAP'01, November 9, 2001, Atlanta, Georgia, USA.  
Copyright 2001 ACM 1-58113-437-1/01/0011...\$5.00.

and compound indexes for materialization for a given query set pattern is a difficult task [10] and depends on the specific query set pattern. Bitmap indexes (and their variants) [16] are very popular because of their compactness and support of star joins. Nevertheless, in many cases, multidimensional access methods (e.g., R-tree) outperform bit-mapped indexing methods [18].

**Contribution** In this paper we focus on heuristically optimizing OLAP queries in databases that are clustered with respect to multiple hierarchical dimensions using multidimensional access methods. The main contributions are the following:

- We present a multidimensional database architecture based on the star model (called CSB star). The dimension tables are organized using one-dimensional hierarchical clustering and encoding techniques, while the fact table is organized using a multidimensional access method.
- We show how OLAP queries can be easily expressed by the users over a traditional star schema. The CSB star schema is intended to be a storage option only. The query processor rewrites user queries over the CSB star schema.
- We exploit the clustering features of the CSB star schema to efficiently process a class of typical OLAP queries. The expensive star-join operations needed in a traditional star schema can be essentially implemented as multidimensional range restrictions on the fact table and range restrictions on the dimension tables. Supplementary joins are implemented as merge join operations on sorted tables. Grouping operations are performed on partially sorted relations.
- In this context we detect special cases where joins of fact table tuples with tuples from the dimensions can be avoided, and the grouping of the tuples can be performed only once before all join operations.

This work is done in the context of the European IST project "EDITH". In this project we use a multidimensional access method integrated into the kernel of a database management system [17].

**Outline** The next section reviews related work. Section 3 introduces the basic concepts of multidimensional hierarchical clustering adopted here. In Section 4, the architecture of the multidimensional database is presented. Section 5 describes the class of queries considered, introduces a number of physical operators, and shows how queries in this class can be heuristically optimized by exploiting the clustering scheme of the multidimensional database architecture. Section 7 contains concluding remarks and directions for further work.

## 2. RELATED WORK

Conventional query optimizers exploit the knowledge about the group-by clause in a query only by including the grouping columns in the list of interesting orders during join enumeration. The group-by operation can be pushed past one or more joins. This early grouping may reduce the query processing cost by reducing the amount of data participating in the joins. Necessary and sufficient conditions for deciding when this transformation is valid are provided in [22]. A generalization of the

early grouping transformation, the coalescing grouping transformations allow us (a) to perform early group-by but require additional group-by subsequently that coalesces multiple groups and (b) to deal with the case where not all the aggregating columns are present in the node of the query evaluation plan where an early group-by operator is placed [2]. Different other cases of early grouping and aggregation are studied and categorized in [23], along with their reverse transformations of lazy grouping and aggregation. These latter transformations postpone the application of a grouping operation until after a join, and may reduce the number of input rows to the group-by, if the join is selective. Both directions of transformation are considered during query optimization. Transformations as well as optimization algorithms for queries with aggregate views and queries containing aggregate nested subqueries are presented in [3]. The proposed pull-up transformation (the equivalent of lazy grouping and aggregation) makes it possible to reorder relations that belong to different query blocks so that these relations can be joined before the group-by operators are applied. The generalized projection operator, an extension of the duplicate eliminating projection operator, captures the semantics of group-by, aggregation, duplicate-eliminating projection and duplicate preserving-projection in a common unifying framework [8]. In this framework query rewriting rules are able to push aggregation operators past selection conditions (and vice-versa). The pull-up transformation does not apply in the evaluation plans that we consider here because the join operations do not reduce the number of tuples of the joining tables. In contrast, a coalescing grouping transformation can be very efficiently exploited: an early grouping can be pushed past all joins. It is worth noting that this is due to the architecture of the CSB star schema that uses hierarchical clustering and encoding techniques, and does not apply to a traditional star join schema. Related works dealing with multidimensional access methods and multidimensional hierarchical clustering are cited in the next section.

## 3. MULTIDIMENSIONAL HIERARCHICAL CLUSTERING

We present in this section the basic concepts of multidimensional hierarchical clustering and range query processing adopted in the paper.

**Multidimensional clustering and the UB-tree.** A tuple of a relation in a relational database can be viewed as a point in a multidimensional space where the dimensions are determined by the attributes of the tuple. In this context, the processing of queries can be supported by multidimensional access methods [6]. OLAP queries often impose restrictions on multiple attributes (dimensions). Multidimensional access methods are used to cluster data with respect to multiple dimensions. Multidimensional clustering can substantially speed up queries that restrict multiple dimensions. The main problem in the design of multidimensional access methods is that there exists no total ordering among the points in the multidimensional space that preserves spatial proximity. One way to heuristically deal with this problem is

to discover a total order that preserves spatial proximity to some extent. This total order is called *space-filling curve*. Then, a one-dimensional access method can be used in combination with the space-filling curve to improve the access of the points in the space. Such a solution to the problem is provided by the *UB-tree* [1].

**Range queries on UB-trees and the Tetris algorithm.** A query that restricts all attributes (dimensions) to an interval is called range query. The multidimensional interval determined by the one-dimensional intervals is called query box. In order to answer a range query, [1] presents an algorithm for UB-trees that fetches from the disk only those regions (pages) that properly intersect the query box. The Tetris algorithm [15] is a generalization of the multidimensional range query algorithm that efficiently combines sort operations with the evaluation of multi-attribute restrictions. The Tetris algorithm takes as input an attribute of a relation and a query box determined by the restrictions of a query on the relation, and returns the tuples of the relation satisfying the restrictions, ordered on the input attribute. Compared to the access methods of commercial systems for queries in the TPC-D benchmark [21], the Tetris algorithm shows significant speedups, important temporary storage requirement reduction for the sorting process, and multiple times faster production of the first results of a sort operation [15].

**Dimension hierarchies.** A dimension hierarchy  $D$  of depth  $k$  is a list  $L^k, \dots, L^1$  of  $k$  names which are called *dimension* or hierarchy levels. With every level  $L$  in  $D$ , a non-empty finite set of values  $dom(L)$  is associated through the function  $dom$  such that  $dom(L^i) \cap dom(L^j) = \emptyset$ ,  $i \neq j$ . In each dimension, we additionally assume an auxiliary level  $L^{k+1}$  whose domain contains the single value *all*. For every two levels  $L^i, L^{i+1}$ ,  $i \in [1, k]$ , a function *parent* from  $dom(L^i)$  onto  $dom(L^{i+1})$  is defined. For every two levels  $L^{i+1}, L^i$ ,  $i \in [1, k]$ , a function *children* from  $dom(L^{i+1})$  to the power set of  $dom(L^i)$  is defined:  $children(v)$  is the set of values  $v'$  in  $dom(L^i)$  such that  $parent(v') = v$ . Clearly, if  $v, v' \in dom(L^i)$ ,  $i > 1$ , and  $v \neq v'$ ,  $children(w) \neq \emptyset$ , and  $children(v) \cap children(v') = \emptyset$ . A value in  $dom(L^i)$ ,  $i > 1$ , represents a set of values in  $dom(L^1)$  through the function *children*. Level  $L^1$  (the lowest level in the hierarchy of dimension  $D$ ) corresponds to the smallest (finest) granularity of viewing data. Level  $L^{k+1}$  (the top level in the hierarchy of dimension  $D$ ) corresponds to the largest granularity of viewing data. A dimension hierarchy defines a hierarchy tree: the nodes of the tree are the values in  $\bigcup_{i=1}^{k+1} dom(L^i)$ , while the edges are determined by the parent function. The leaf nodes of the tree are the values in  $dom(L^1)$ . The root node is the unique value of  $dom(L^{k+1})$ . The *path* of a node (value)  $n^i \in dom(L^i)$ ,  $i \in [1, k]$ , is defined to be the concatenation of the nodes  $n^k, n^{k-1}, \dots, n^1$  on the path in a hierarchy tree from the root node to  $n^i$ .

**Hierarchy clustering and encoding.** OLAP queries impose restrictions on different levels of the dimension hierarchies. Hierarchy clustering and encoding [24, 14] in combination with multidimensional access methods can be used to speed up these queries and to optimize

the storage usage.

In order to take into account dimension hierarchies in the clustering of data, instead of the values  $v$  of the lowest level of a dimension, their path  $p$  in the hierarchy tree of the dimension is considered. The concatenated values can be shortened using the following encoding schema which is quite similar to that of [14]. Let  $v$  be a value in the domain of level  $L^i$ ,  $i \in [1, k]$ . Let also  $v'$  be the parent of  $v$  in the hierarchy tree,  $V$  be the cardinality of  $children(v')$ , and  $<_q$  be the "less than" comparison operator of the query language. We define a one-to-one function  $S : children(v') \rightarrow [0, V - 1]$  such that: for every  $u, u' \in children(v')$ ,  $u <_q u'$  implies  $S(u) < S(u')$ .  $S(w)$  is called the *surrogate* of  $v$ . Note that if  $<_q$  is not defined in the query language for the values in  $dom(L^i)$ ,  $S$  is simply defined as a one-to-one function from  $children(v')$  onto  $[0, V - 1]$ .

Let  $v$  be a value in  $\bigcup_{i=1}^k dom(L^i)$ . Then, the compound surrogate of  $v$ ,  $C(w)$ , is the path of  $v$  where the concatenated values are replaced by their surrogates. One way to compactly store compound surrogates is as fixed length strings of bits: for each level  $L^i$ ,  $i \in [1, k]$ , in the hierarchy, the maximum surrogate  $x$  is defined as  $\max\{V - V$  is the cardinality of  $children(w)$  and  $v \in dom(L^{i-1})\}$ . Then, at least  $\lceil \log_2 x \rceil$  bits are reserved in the binary representation of the compound surrogate for the surrogates of level  $L^i$ . The total number of bits reserved for a level  $L^i$  is called *spread* of  $L^i$ .

#### 4. THE CSB STAR SCHEMA

We present in this section the architecture of our multidimensional database. The schema of the multidimensional database is a star with one fact table  $F$  and dimension tables  $D_1, \dots, D_n$ .

**The dimension tables.** The schema of a dimension table  $D_i$  corresponding to a dimension hierarchy  $D_i$  of depth  $k_i$  consists of:

- A set  $H_i$  of *hierarchy attributes*  $\{H_i^1, H_i^2, \dots, H_i^{k_i}\}$  that correspond one-to-one to the  $k_i$  levels of the dimension hierarchy;  $H_i^1$  corresponds to the lowest level in the hierarchy and  $H_i^{k_i}$  to the highest.
- A set  $F_i$  of *feature attributes* that provide descriptive characterizations of the different levels of the dimension. A feature attribute  $F_i^j$  characterizes the hierarchy attribute  $H_i^j$ . Feature attributes are optional in the schema of a dimension table.
- A compound surrogate attribute  $C_i$ .

If  $t$  is a tuple in table  $D_i$ ,  $t[H_i^j] \in dom(H_i^j)$ ,  $j \in [1, k_i]$ , and  $parent(t[H_i^j]) = t[H_i^{j+1}]$ ,  $j \in [1, k_i - 1]$ .  $t[C_i]$  is the compound surrogate of  $t[H_i^1]$ .

By the definition of the compound surrogate of a value, it is clear that a point restriction on a hierarchy attribute of a dimension table can be expressed as a single range restriction on the compound attribute of this dimension table.

The compound surrogate attribute  $C_i$  is the primary key of the dimension table  $D_i$ . By the definition of a dimension hierarchy,  $H_i^1$  is also a key of table  $D_i$ , and the following functional dependencies between hierarchy attributes hold on  $D_i$ :  $H_i^j \rightarrow H_i^{j+1}$ ,  $j \in [1, k_i - 1]$ .

1]. Furthermore, the following functional dependencies between hierarchy and features attributes hold on  $D_i$ :  $H_i^j \rightarrow F_i^j$ ,  $i \in [1, k_i]$ , where  $F_i^j$  is a feature attribute characterizing the hierarchy attribute  $H_i^j$ .

We associate with a dimension table a primary index  $P_i$  on  $C_i$ , and a secondary (compound) index  $I_i$  on  $H_i^{k_i}, \dots, H_i^1, C_i$ . Index  $I_i$  is important for computing ranges of values on  $C_i$  from point or range restrictions on the different hierarchy attributes of  $D_i$ .

Since the tuples of  $D_i$  are clustered on  $C_i$ , range restrictions on  $C_i$  can be computed efficiently. This clustering provides also a grouping of the tuples of the dimension table with respect to any hierarchy attribute. This property is particularly useful for evaluating grouping/aggregation queries as those that are extensively used in OLAP applications.

**The fact table.** The schema of the fact table  $F$  consists of:

- (a) The compound surrogate attributes  $C_1, \dots, C_n$ , one for each dimension table  $D_i$ ,  $i \in [1, n]$ .
- (b) A set of measure attributes  $M = \{M_1, \dots, M_k\}$ .

The set of attributes  $C_1, \dots, C_n$  is the primary key of  $F$ . Each  $C_i$  in  $F$  is a foreign key and refers to attribute  $C_i$  of the dimension table  $D_i$ . In general, a fact table contains a huge number of tuples. In a traditional star schema, the lower level hierarchy attributes are stored as foreign keys in the fact table. Instead, by storing the compound surrogate attributes as foreign keys in the fact table, we importantly reduce its size. Table  $F$  is organized as a UB-tree on the attributes  $C_1, \dots, C_n$ . The schema of our multidimensional database is called Compound Surrogate Based star schema (CSB star for short).

**User View.** The clustering scheme of the CSB star schema is intended to be a storage option only, without affecting the formulation of queries by the user. User queries are easily formulated on a simple star schema (called user star schema) as this is defined by the view  $FT$  for the fact table, and the views  $DT_1, \dots, DT_n$  for the  $n$  dimension tables:

```
CREATE VIEW FT AS
SELECT  H_1^1, \dots, H_n^1, M
FROM    F, D_1, \dots, D_n
WHERE   F.C_1 = D_1.C_1, \dots, F.C_n = D_n.C_n
```

```
CREATE VIEW DT_i AS
SELECT  H_i, F_i
FROM    D_i
```

Clearly, the views  $FT$  and  $DT_1, \dots, DT_n$  form a typical star schema. User queries formulated on the user schema are rewritten by the query processor over the tables  $F$  and  $D_1, \dots, D_n$  by replacing the views by their view definitions. In the following we consider queries that are rewritten over the CSB star schema.

## 5. OPTIMIZING OLAP QUERIES

We show in this section how to heuristically optimize OLAP queries by exploiting the clustering scheme and the access methods of the multidimensional database architecture.

### 5.1 The class of queries considered

We consider OLAP queries of the form shown below. Typical cases of OLAP operations can be expressed by this SQL query.

```
SELECT  X, A
FROM    F, D_1, \dots, D_n
WHERE   c^J AND c^H AND c^F
GROUP BY G
HAVING  c^A
ORDER BY O
```

$X$  is a set of hierarchy and/or feature attributes (called projected grouping attributes).  $A$  is a set of aggregated measures (aggregate functions on measure attributes). Using the categorization of aggregate functions introduced in [7], we focus on distributive SQL aggregate functions: min, max, sum, count.  $G$  is a set of hierarchy and/or feature attributes (called *grouping attributes*).  $c^J$  is a conjunction of equi-joining conditions on the compound surrogates.  $c^H$  is a conjunction of comparisons involving exclusively hierarchy attributes.  $c^F$  is a conjunction of comparisons involving exclusively feature attributes;  $c^A$  is a conjunction of comparisons involving exclusively aggregated measures from  $A$ .  $O$  is a list of attributes from  $X \cup A$ . The joining conditions are of the form  $F.C_i = D_i.C_i$ . The comparisons involving an attribute  $A$  are of the form  $A \theta c$  where  $\theta$  is one of the comparison operators  $<, \leq, =, \geq, >$ , and  $c$  is a constant value.

We assume that at least one hierarchy attribute from each dimension is involved in a comparison in the WHERE clause of the query. We also assume that if comparisons of the form  $H_i^j \theta c$ , where  $\theta \in \{<, \leq, \geq, >\}$  appear in the WHERE clause of a query, then the parent function of the dimension hierarchy  $D_i$  is *monotone*. A parent function is monotone if for every  $v, v' \in \text{dom}(H_i^j)$ ,  $j = 1, \dots, k_i$ , if  $v \leq v'$  then  $\text{parent}(w) \leq \text{parent}(v')$ . This assumption guarantees that a range restriction on a hierarchy attribute of a dimension table can be expressed as a single range restriction on the compound surrogate of the table. In this section we consider queries whose hierarchy and feature attribute restrictions can be expressed as a multidimensional range restriction on the compound surrogate attributes of all the dimensions tables (i.e., a single query box).

### 5.2 Physical operators

In order to construct an OLAP query evaluation plan, we use a number of physical operators that are presented below. Some of them are the traditional relational operators and the others are specific to the organization of the multidimensional database.

By abuse of notation, we view a compound surrogate attribute  $C_i$  as a composite attribute consisting of surrogate attributes  $S_i^{k_i}, \dots, S_i^1$ . If  $t$  is a tuple in the dimension table  $D_i$ ,  $t[S_i^j]$ ,  $j \in [1, k_i]$ , is the binary representation of the surrogate of  $t[H_i^j]$  ( $t[S_i^j] = S(t[H_i^j])$ ).  $t[S_i^j]$  is part of  $t[C_i]$  and its length in bits is equal to the spread of attribute  $H_i^j$ .

We denote by  $P_X$  the projection with duplicate retention operator on the set of attributes  $X$ , and by  $\Pi_X$  the

set-theoretic projection operator (SELECT DISTINCT) on  $X$ .  $\sigma_c$  denotes the selection operator with selection condition  $c$ .  $S[Y]$  denotes the sorting operator on the list of attributes  $Y$ .  $\bowtie_Y$  denotes the natural join operator on the list of attributes  $Y$ . The natural join operator is applied on tables that are sorted on the list of attributes  $Y$  and is implemented as merge join.  $\Pi_{X,A}$  denotes the generalized projection operator [8] (grouping/aggregation operator), where  $X$  is a set of grouping attributes and  $A$  is a set of aggregate functions on measure attributes.

The *Tetris* operator, denoted  $T[l_1, u_1], \dots, [l_n, u_n]; C_i$ ,  $i \in [1, n]$ , can be applied to the fact table and represents the Tetris algorithm.  $[l_j, u_j]$ ,  $j \in [1, n]$ , is a range of values for the compound surrogate attribute  $C_j$ , and  $C_i$  is the compound surrogate attribute on which the resulting table is ordered (refer to Section 3). The schema of the resulting table is that of the fact table.

The *Range* operator, denoted  $R[c_i^H; B]$  can be applied to the compound index  $I_i$  of a dimension table  $D_i$ .  $c_i^H$  is a condition of the type described in Subsection 5.1, and involves hierarchy attributes of dimension table  $D_i$ .  $B$  is a boolean variable that takes values '1' and '0'.  $R[c_i^H; 0]$  on  $I_i$  returns a range of values  $[l_i, u_i]$  on  $C_i$  such that restricting  $C_i$  of  $D_i$  in this range is equivalent to applying the restriction  $c_i^H$  to  $D_i$ .  $R[c_i^H; 1]$  on  $I_i$  returns, besides  $[l_i, u_i]$ , a table  $T$ . The schema of  $T$  is  $H_i^1, \dots, H_i^k, C_i$  and its content is the set of tuples  $t$  over  $H_i^1, \dots, H_i^k, C_i$  in  $I_i$  such that  $l_i \leq t[C_i] \leq u_i$ .

### 5.3 Construction of the evaluation plan

We show now how to optimize OLAP queries of the type presented in Subsection 5.1. Our approach is heuristic and is not based on a specific cost model. For ease of presentation we use an example query that is general enough to encompass different optimization cases, and we show how our technique can be applied to produce an evaluation plan.

**EXAMPLE 5.1.** We consider the following query defined over a four dimensional schema.

```
SELECT   F13, H22, H33, F33, sum(M)
FROM     F, D1, D2, D3, D4
WHERE    F.C1 = D1.C1 AND F.C2 = D2.C2 AND
         F.C3 = D3.C3 AND F.C4 = D4.C4 AND
         c1H AND c2H AND c3H AND c4H AND c3F
GROUP BY F13, H22, H33, F33
HAVING   c1A
ORDER BY H33, H22
```

A query evaluation plan for this query that takes advantage of our multidimensional database architecture is shown in Figure 1. The expensive star-join operations required in a traditional star schema are here essentially implemented by a multidimensional range restriction on the fact table [14]. The presence of the compound surrogate attributes in the fact table allows for an early grouping and aggregation operation. This operation is facilitated by the fact that the selected fact table tuples are retrieved sorted on a compound surrogate attribute. The evaluation plan comprises two kinds of nodes: operation nodes representing operations, and data nodes

representing input data, and intermediate and final results. An operation node is depicted by a small circle and is labeled by the operation(s) it represents. Some operations may be pipelined in which case the corresponding temporary results are not actually stored on the disk. The fact table and the dimension tables are depicted by bigger circles, the compound indexes by rectangles, and the ranges of computed compound surrogate attribute values by triangles. Before discussing, in the following, the different steps of the evaluation plan, we provide some definitions.

**DEFINITION 5.1.** A (hierarchy or feature) attribute is called restricted attribute if it is involved in a selection condition ( $c''$  or  $c'''$ ) in the query.

An attribute is called imported attribute if it is a projected grouping hierarchy attribute or a grouping feature attribute in the query. A dimension containing imported attributes is called *joining dimension*.  $\square$

An imported attribute needs to be added to selected fact table tuples. A grouping hierarchy attribute that is not projected in the query need not be added to the fact table tuples since the compound surrogate attributes can be used for performing the grouping operation. For each joining dimension, a join operation is needed in order to add the imported attributes of the dimension to selected fact table tuples.

**DEFINITION 5.2.** A joining dimension  $D_i$  is called candidate last joining dimension if one of the following two conditions hold:

- The first attribute in the list of sorting attributes in the query (the list 0 of attributes in the ORDER BY clause of the query) is a hierarchy attribute of  $D_i$ .
- The first attribute in the list of sorting attributes in the query is not a hierarchy attribute (or there is no such a list), there is a grouping feature attribute in the query, and there is a grouping hierarchy attribute of  $D_i$  in the query.  $\square$

Selecting the dimension involved in the last join operation to be a candidate last joining dimension simplifies the final grouping and sorting operations.

#### Computing ranges of compound surrogate values

The first step in the construction of the query evaluation plan is the application of the Range operator  $R[c_i^H; B]$  to the compound index  $I_i$  of each dimension.  $R[c_i^H; B]$  computes the lower and the upper bound of the range of values  $[l_i, u_i]$  of the compound surrogate using the comparisons in  $c_i^H$  and the compound index  $I_i$ . The range  $[l_i, u_i]$  is always provided as input to the Tetris operator. It can also be used for a selection operation on the dimension table  $D_i$ . If only hierarchy (and no feature) attributes from a dimension are involved in a query, then their values can be obtained directly from the compound index  $I_i$ , without accessing the dimension table  $D_i$ . In this case the parameter  $B$  of  $R[c_i^H; B]$  is set to '1'. In general the application of the range operator follows the rules below:

- If there is no imported attribute or restricted feature attribute of the dimension in the query then

$B$  is set to '0' and the computed range is used only by the Tetris operator.

- (b) If the imported attributes of the dimension are only hierarchy attributes and there is no restricted feature attribute of the dimension in the query then  $B$  is set to '1'. The computed range is provided to the Tetris algorithm only, while the set of tuples retrieved from  $I_i$  are used in a subsequent join operation.
- (c) If the imported attributes of the dimension include a feature attribute or there is a restricted feature attribute of the dimension in the query then  $B$  is set to '0'. The computed range is provided both to the Tetris algorithm and to a selection operation on the corresponding dimension table.

**Restricting the dimension tables** If the imported attributes of the dimension include a feature attribute or there is a restricted feature attribute of the dimension in the query, the dimension table needs to be accessed. The primary index on the compound surrogate attribute  $C_i$  of the dimension is used to retrieve the tuples of the dimension table that fall within the range of values computed by the range operator. Those of these tuples that satisfy the condition  $c_i^F$  are retained projected over an appropriate set of attributes. If this computation returns an empty set of tuples, the whole processing of the query is ended since the answer is an empty set. Otherwise, the computed tuples are subsequently joined with tuples derived from the fact table. The set  $S$  of attributes of  $D_i$  to be projected are determined as follows:

- (a)  $S$  includes all the imported attributes of  $D_i$ , and
- (b) From the surrogate attributes  $S_i^{k_i}, \dots, S_i^1$  of the compound surrogate  $C_i$ ,  $S$  includes the surrogate attributes  $S_i^{k_i}, \dots, S_i^j$ , where  $j$  is the minimal level of the imported attributes of  $D_i$ .

If the minimal level  $j$  is high enough in the dimension hierarchy this duplicate elimination projection is expected to significantly reduce the number of selected tuples.

It is important to note that the tuples from the dimension table are retrieved sorted on the compound surrogate attribute  $C_i$ . Since  $C_i$  is the concatenation of the surrogate attributes  $S_i^{k_i}, \dots, S_i^1$ , these tuples are also sorted with respect to the list of surrogate attributes  $S_i^{k_i}, \dots, S_i^j$ . As a consequence, the elimination of duplicates required for the projection operation can be performed without extra cost. Furthermore, the sort order of the output tuples can be exploited in a subsequent merge join operation with tuples derived from the fact table.

**Multidimensional range selection and sorting** The Tetris operator takes the ranges of values on the compound surrogate attribute computed in the first step, and retrieves from the fact table the qualifying tuples sorted on a compound surrogate attribute  $C_i$ . In general, the choice of the sorting attribute  $C_i$  does not affect the performance of the Tetris operator. We assume that the cache memory requirements of the Tetris algorithm are satisfied by the available main memory [15]. The sorting attribute  $C_i$  for the Tetris algorithm is chosen from a dimension according to the following rules:

- (a)  $C_i$  is chosen from a joining dimension that is not a candidate last joining dimension.
- (b) If  $C_i$  cannot be chosen by the rule (a), it is chosen from a joining dimension.
- (c) If  $C_i$  cannot be chosen by rules (a) or (b), it is chosen from a dimension that has a hierarchy attribute involved as a grouping attribute in the query.
- (d) If  $C_i$  cannot be chosen by the previous rules, it is chosen arbitrarily.

**Processing of the selected fact table tuples** The presence of the compound surrogate attributes  $C_1, \dots, C_n$  in the fact table, allows for an early grouping and aggregation of the fact table tuples resulting by the Tetris algorithm [5]. This operation can be performed efficiently due to the fact that the tuples resulting by the Tetris operation are already sorted (and thus grouped) with respect to one compound surrogate attribute  $C_i$ ,  $i \in [1, n]$ . The set of attributes on which the grouping is performed comprises the surrogate attributes  $S_i^{k_i}, \dots, S_i^j$  from each dimension  $D_i$  that contains a grouping attribute in the query.  $j$  is the minimal level of the (hierarchy and feature) grouping attributes of  $D_i$  in the query. Usually, in OLAP applications, a fact table contains a huge number of tuples which are grouped to produce a small number of aggregated results. Therefore, this early grouping operation is expected to drastically reduce the number of fact table tuples at an early stage of their processing.

If there is no grouping feature attribute in the query then the final grouping and aggregation operation that will be presented in the next step is not needed. The reason is that the compound surrogate attributes have already been used instead, for grouping on hierarchy attributes. In this case the selection operation on the aggregated measures ( $\sigma_{CA}$ ) can follow immediately after the early grouping operation to further reduce the number of aggregated fact table tuples that are left to be processed.

If there is at least one joining dimension, one of them has provided its compound surrogate attribute as a sorting attribute to the Tetris operator. In this case, the aggregated tuples are equi-joined on the common attributes with the selected tuples of this dimension. Since both sets of tuples are sorted on their common attributes, a merge join algorithm can be efficiently applied. Each aggregated tuple joins with exactly one tuple from the dimension table. This operation does not alter the number of tuples resulting by the grouping/aggregation operation. It does add to these tuples the imported attributes of the joining dimension. Grouping hierarchy attributes of the dimension table that are not projected grouping attributes in the query are not needed since the fact table tuples have already been grouped using the surrogate attributes.

A similar operation is performed on the resulting tuples for each other joining dimension. This operation has to be preceded by a sort operation with respect to the joining attributes, and by a project operation that eliminates the attributes not needed in subsequent operations. The sort operation has to be performed only on the tuples resulting from the fact table since the di-

mension tuples are already sorted with respect to the joining attributes. The order of the join operations does not significantly matter. The only rule that has to be respected is to choose a candidate last joining dimension for the final join operation. This way the sort order of the resulting tuples can be exploited in the subsequent operations.

**Final grouping/aggregation and sorting** In the last step of the evaluation plan the tuples resulting by the last join operation are grouped and aggregated. The aggregated tuples that satisfy the condition  $c^A$  on aggregated measures are retained projected over the attributes required in the output. As mentioned previously, these operations can be avoided when there is no grouping feature attribute in the query. If the output tuples are required sorted, a final sorting operation is also performed. These operations exploit the sort order of the tuples resulting from the previous step.

## 6. CONCLUSION

OLAP applications view data structured in multiple hierarchically organized dimensions. Complex grouping/aggregation queries for OLAP applications process enormous quantities of data and require fast response time. Recent research suggests that multidimensional access methods outperform one-dimensional indexing techniques.

We have presented the CSB star, an architecture for a multidimensional database that is based on the star schema. This architecture uses one-dimensional hierarchical clustering and encoding techniques to organize the dimension tables and multidimensional access methods to organize the fact table. Users can express their queries over a traditional star schema, which are then rewritten by the query processor over a CSB star schema. We have shown how the features of this schema allow the heuristic optimization of a class of typical OLAP queries: expensive star-join operations are essentially reduced to multidimensional and one-dimensional range restrictions, supplementary joins are implemented as merge join operation on sorted tables, and grouping operations are performed on partially sorted data. We have detected special cases where supplementary joins are avoided, and a grouping operation can be pushed past all join operations.

An interesting extension of the present work concerns considering a larger class of queries. In particular relaxing a number of the restrictions adopted here can result in queries that determine multiple query boxes on the compound surrogate attributes. Our results apply to this case too by considering the different query boxes separately. However, a cost based optimization technique that considers different groupings of these query boxes is expected to provide further improvements.

## 7. REFERENCES

- [1] R. Bayer. The Universal B-Tree for Multidimensional Indexing: General Concepts. In *Proc. of the Intl. Conf. on World-Wide Computing and its Applications*, Springer, LNCS 1274, pages 98-112, 1997.
- [2] S. Chaudhuri and K. Shim. Including Group-By in Query Optimization. In *Proc. of the 20th Intl. Conf. on Very Large Data Bases*, pages 354-366, 1994.
- [3] S. Chaudhuri and K. Shim. Optimizing Queries with Aggregate Views. In *Proc. of the 5th Intl. Conf. on Extending Database Technology*, pages 167-182, 1996.
- [4] S. Dar, H. V. Jagadish, A. Y. Levy, and D. Srivastava. Answering SQL Queries with Aggregation using Views. In *Proc. of the 22nd Intl. Conf. on Very Large Data Bases*, pages 318-329, 1996.
- [5] K. Elhardt. EDITH query processing. Technical report, Transaction Software, Jan 2001.
- [6] V. Gaede and O. Günther. Multidimensional Access Methods. *ACM Computing Surveys*, 30(2), 1997.
- [7] J. Gray, A. Bosworth, A. Layman, and H. Pirahesh. Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals. In *Proc. of the 12th ICDE*, 1996.
- [8] A. Gupta, V. Harinarayan, and D. Quass. Aggregate-Query Processing in Data Warehousing Environments. In *Proc. of the 21st Intl. Conf. on Very Large Data Bases*, pages 358-369, 1995.
- [9] A. Gupta and I. S. Mumick. Maintenance of materialized views: Problems, techniques and applications. *Data Engineering*, 18(2):3-18, 1995.
- [10] H. Gupta, V. Harinarayan, A. Rajaraman, and J. D. Ullman. Index Selection for OLAP. In *Proc. of the 13th Intl. Conf. on Data Engineering*, pages 208-219, 1997.
- [11] V. Harinarayan, A. Rajaraman, and J. D. Ullman. Implementing Data Cubes Efficiently. In *Proc. of the ACM SIGMOD Intl. Conf.*, 1996.
- [12] R. Kimball. *The Data Warehouse Toolkit*. John Wiley & Sons, 1996.
- [13] Y. Kotidis and N. Roussopoulos. DynaMat: A Dynamic View Management System for Data Warehouses. In *Proc. of the ACM SIGMOD Intl. Conf. on Management of Data*, pages 371-382, 1999.
- [14] V. Markl, F. Ramsak, and R. Bayer. Improving OLAP Performance by Multidimensional Hierarchical Clustering. In *Proc. of the Intl. Database Engineering and Applications Symposium*, pages 165-177, 1999.
- [15] V. Markl, M. Zirkel, and R. Bayer. Processing Operations with Restrictions in RDBMS without External Sorting: The Tetris Algorithm. In *Proc. of the 15th Intl. Conf. on Data Engineering*, pages 562-571, 1999.
- [16] P. O'Neil and D. Quass. Improved Query Performance with Variant Indexes. In *Proc. of the ACM SIGMOD Intl. Conf.*, pages 38-49, 1997.
- [17] F. Ramsak, V. Markl, R. Fenk, M. Zirkel, K. Elhardt, and R. Bayer. Integrating the UH-Tree into a Database System Kernel. In *Proc. of the 26th Intl. Conf. on Very Large Data Bases*, pages 263-272, 2000.
- [18] S. Sarawagi. Indexing OLAP Data. *Data Engineering*, 20(1):36-43, 1997.
- [19] D. Theodoratos and T. Sellis. Data Warehouse Configuration. In *Proc. of the 23rd Intl. Conf. on Very Large Data Bases*, pages 126-135, 1997.
- [20] D. Theodoratos and T. Sellis. Answering Queries on Cubes Using Other Cubes. In *Proc. of the Intl. Conference on Scientific and Statistical Databases*, pages 109-122, 2000.
- [21] TPC benchmark D. Technical report, Transaction Processing Performance council, 1997.
- [22] W. Yan and P.-A. Larson. Performing Group-By before Join. In *Proc. of the 10th Intl. Conf. on Data Engineering*, pages 89-100, 1994.
- [23] W. Yan and P.-A. Larson. Eager Aggregation and Lazy Aggregation. In *Proc. of the 21st Intl. Conf. on Very Large Data Bases*, pages 1-13, 1995.
- [24] C. Zou, B. Salzberg, and R. Ladin. Back to the future: Dynamic hierarchical clustering. In *Proc. of the 14th Intl. Conf. on Data Engineering*, pages 578-587, 1998.



